

2/5/1 (Item 1 from file: 347)
DIALOG(R) File 347:JAPIO
(c) 2003 JPO & JAPIO. All rts. reserv.

04315811 **Image available**
PROTOCOL SIMULATION DEVICE

PUB. NO.: 05-307511 [JP 5307511 A]
PUBLISHED: November 19, 1993 (19931119)
INVENTOR(s): KANAJI KATSUYUKI
KAGAYA SATOSHI
APPLICANT(s): TOSHIBA CORP [000307] (A Japanese Company or Corporation), JP
(Japan)
APPL. NO.: 04-111538 [JP 92111538]
FILED: April 30, 1992 (19920430)
INTL CLASS: [5] G06F-013/00; G06F-009/46; G06F-011/28
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- Memory Units); 45.1
(INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1700, Vol. 18, No. 121, Pg. 77,
February 25, 1994 (19940225)

ABSTRACT

PURPOSE: To automatically execute the dynamic verification of a protocol without necessitating a special practical environment or a simulation environment.

CONSTITUTION: This device is provided with a simulation environment setting part 10 for setting a simulation environment required for executing the dynamic verification of a communication protocol, through an input from a user, a simulation executing part 11 for executing the dynamic verification of the communication protocol on the set simulation environment, based on a state transition diagram, and a display control part 12 for displaying the present simulation state, that is, the state transition diagram and the simulation environment of the protocol being in the course of simulation at present, and moreover, the generation part of a protocol error and the error contents, etc., so as to be easily understandable to a user.

Best Available Copy

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-307511

(43)公開日 平成5年(1993)11月19日

(51)Int.Cl.⁵

G 0 6 F 13/00

9/46

11/28

識別記号

3 5 3 U

3 3 0 C

3 4 0 C

庁内整理番号

7368-5B

8120-5B

9290-5B

F I

技術表示箇所

審査請求 未請求 請求項の数 5 (全 22 頁)

(21)出願番号

特願平4-111538

(22)出願日

平成4年(1992)4月30日

(71)出願人

000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者

金地 克之

神奈川県川崎市幸区柳町70番地 株式会社
東芝柳町工場内

(72)発明者

加賀谷 聡

神奈川県川崎市幸区柳町70番地 株式会社
東芝柳町工場内

(74)代理人

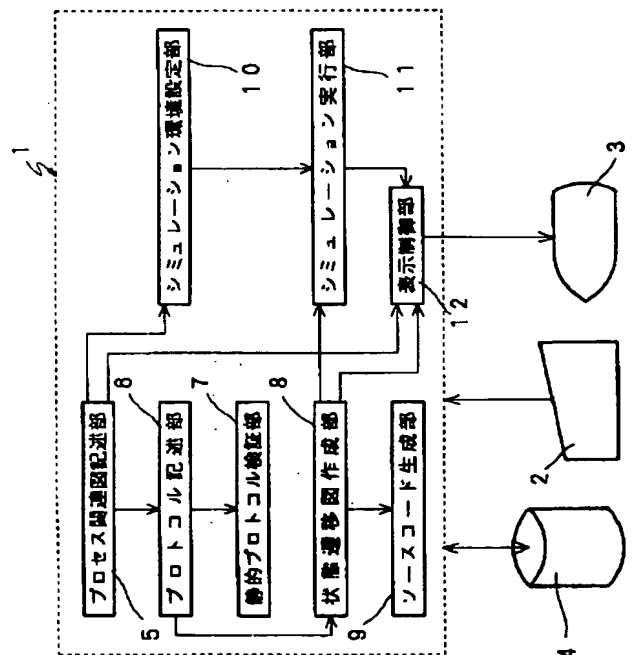
弁理士 須山 佐一

(54)【発明の名称】 プロトコル・シミュレーション装置

(57)【要約】

【目的】 プロトコルの動的な検証を特別な実地環境あるいは模擬環境を要することなく自動的に行うことを目的としている。

【構成】 通信プロトコルの動的な検証を行う上で必要なシミュレーション環境をユーザからの入力を通じて設定するシミュレーション環境設定部10と、設定されたシミュレーション環境上で通信プロトコルの動的な検証を状態遷移図に基づいて実行するシミュレーション実行部11と、現在のシミュレーション状況、すなわち現在シミュレーション中のプロトコルの状態遷移図やシミュレーション環境、さらにはプロトコルエラーの発生箇所およびエラー内容等をユーザに分かりやすく表示するための制御を行う表示制御部12とを有する。



【特許請求の範囲】

【請求項1】 プロセス間の通信手順を示す通信プロトコルの検証を行うプロトコル・シミュレーション装置において、

前記通信プロトコルに関する各プロセスの状態遷移図を作成する状態遷移図作成手段と、

少なくとも前記プロセス間のチャネルの通信容量および前記チャネルの信号伝送時間を含むシミュレーション環境を設定する設定手段と、

前記状態遷移図作成手段により作成された状態遷移図に基づき、前記設定手段にて設定されたシミュレーション環境上で前記通信プロトコルの動作をシミュレーションし、かつシミュレーション中に発生するエラーを検出する検証手段と、

前記検証手段の検証結果を表示する表示手段とを有することを特徴とするプロトコル・シミュレーション装置。

【請求項2】 請求項1記載のプロトコル・シミュレーション装置において、

前記設定手段は、少なくとも前記プロセス間のチャネルの通信容量、前記チャネルの信号伝送時間、前記プロセス上のタスク処理時間、および前記プロセスを永続的な待ち状態から保護するためのタイムアウト時間を含むシミュレーション環境を設定することを特徴とするプロトコル・シミュレーション装置。

【請求項3】 請求項1または2記載のプロトコル・シミュレーション装置において、

前記状態遷移図は少なくともプロセス上の状態、信号受信、信号送信およびタスク処理をそれぞれ固有のシンボルを用いて表現して構成され、前記検証手段はこの状態遷移図のシンボルを解釈しつつこのシンボルに対応するイベントの処理を実行することによりシミュレーションを実行することを特徴とするプロトコル・シミュレーション装置。

【請求項4】 請求項2または3記載のプロトコル・シミュレーション装置において、

前記検証手段は、前記状態遷移図に基づくタイマー設定の発生を検出して起動されるタイマー手段を有し、このタイマー手段のタイマー時間が前記シミュレーション環境上で設定されたタイムアウト時間を越えたとき非実行遷移状態としてエラーを判定することを特徴とするプロトコル・シミュレーション装置。

【請求項5】 請求項1乃至3記載のいずれかのプロトコル・シミュレーション装置において、

前記検証手段は、前記チャネルに送信された信号の数が前記チャネルの通信容量を越えたときチャネルオーバーフローとしてエラーを判定することを特徴とするプロトコル・シミュレーション装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、通信ソフトウェア開発

において設計された通信プロトコルのシミュレーションを行うプロトコル・シミュレーション装置に関する。

【0002】

【従来の技術】 従来、通信ソフトウェア開発におけるプロトコル（通信システムの動作実態であるプロセス間の通信手順）の検証では、デッドロックや受信不能状態等の障害検出等についてはプロトコル記述の静的な解析により可能である。

【0003】 しかしながら、プロセス内でのタスク処理時間や、チャネルを通じて信号が送られてくる場合等、伝送時間を考慮したプロトコルの動的な検証については、プロトコル設計およびコーディングの完了後、実システムあるいは模擬システムを構築して実地試験を行う必要があり、時間的、労力的に多大なコストがこの動的検証のために費やされていた。また、この場合、プロトコルに誤りがあった場合にプロトコルを根本的に設計し直さねばならないこともあり、これが一連のプロトコル設計作業を複雑にしている要因の一つとなっていた。

【0004】

【発明が解決しようとする課題】 このように従来、通信プロトコルの動的な検証つまり時間的な要素を組み込んだ検証には様々な障害が伴い、効率良くシミュレーションを行うことが困難とされていた。

【0005】 本発明はこのような課題を解決するためのもので、プロトコルの動的な検証を特別な実地環境あるいは模擬環境を要することなく自動的に行うことができ、プロトコル検証作業の効率を大幅に向上させることのできるプロトコル・シミュレーション装置の提供を目的としている。

【0006】

【課題を解決するための手段】 本発明のプロトコル・シミュレーション装置は上記した目的を達成するために、プロセス間の通信手順を示す通信プロトコルの検証を行うプロトコル・シミュレーション装置において、前記通信プロトコルに関する各プロセスの状態遷移図を作成する状態遷移図作成手段と、少なくとも前記プロセス間のチャネルの通信容量および前記チャネルの信号伝送時間を含むシミュレーション環境を設定する設定手段と、前記状態遷移図作成手段により作成された状態遷移図に基づき、前記設定手段にて設定されたシミュレーション環境上で前記通信プロトコルの動作をシミュレーションし、かつシミュレーション中に発生するエラーを検出する検証手段と、前記検証手段の検証結果を表示する表示手段とを有している。

【0007】

【作用】 本発明では、予め設定手段にて、例えばプロセス間のチャネルの通信容量、チャネル間の信号の伝送時間、プロセス上のタスク処理時間、タイムアウト時間等のシミュレーション環境を設定しておく。検証手段はこのシミュレーション環境上で、通信プロトコルの動作を

3

状態遷移図に従ってシュミレーションし、シュミレーション中に発生するエラー、例えば非実行遷移状態やチャネルオーバーフロー等を検出し、その検証結果を表示手段を通して表示する。

【0008】したがって、本発明によれば、特別な実地環境あるいは模擬環境を要することなく自動的にプロトコルの動的な検証を行うことができ、プロトコル検証作業の効率を大幅に向上させることができる。

【0009】

【実施例】以下、本発明の実施例の詳細を図面に基づいて説明する。

【0010】図1は本発明に係る一実施例のプロトコル・シミュレーション装置のシステム構成を示す図である。

【0011】同図において、1は装置本体、2はキーボード等の入力装置、3はCRT等の表示装置、4はフロッピーディスク装置等の外部記憶装置（データベース）である。また装置本体において、5はプロセス関連図の作成をユーザからの入力を通じて行うプロセス関連図記述部である。6は作成されたプロセス関連図に基づくプロトコル設計つまりプロトコル記述の作成をユーザからの入力を通じて行うプロトコル記述部である。7はプロトコル記述に基づく通信プロトコルの静的な検証を自動的に行う静的プロトコル検証部である。8は設計されたプロトコルに対応する状態遷移図を自動合成する状態遷移図作成部である。9は合成された状態遷移図からソースコードを自動生成するソースコード生成部である。また10は通信プロトコルの動的な検証を行う上で必要なシミュレーション環境をユーザからの入力を通じて設定するシミュレーション環境設定部である。11は設定されたシュミレーション環境上で通信プロトコルの動的な検証を状態遷移図に基づいて実行するシュミレーション実行部である。12は現在のシュミレーション状況、すなわち現在シュミレーション中のプロトコルの状態遷移図やシミュレーション環境、さらにはプロトコルエラーの発生箇所およびエラー内容等をユーザに分かりやすく表示するための制御を行う表示制御部である。

【0012】図2はプロセス関連図記述部5により記述された通信システムのプロセス関連図の例を示している。このプロセス関連図において、丸印がプロセス、プロセス間に結ばれた矢印がチャネルを表している。

【0013】図3はこのプロセス関連図に対してシミュレーション環境設定部10で設定したシミュレーション環境を示す図である。ここでシミュレーション環境としては、チャネルの通信容量（capacity）、チャネル間の信号伝送時間（timer）、1タスク処理設定時間 $T(=0.01s)$ 等が設定される。

【0014】また、これらの環境設定内容と共にこのシミュレーション環境画面には、シミュレーションの全体タスク処理時間（ T_a ）が表示され、かつプロセス・マ

4

ネージャーPMとして個々のプロセスのタスク処理時間（ptime）、イベントの待ち状態（wait）およびタイマー時間（図示せず）とが表示されるようになっている。これらの内容はシミュレーションの過程でシュミレーション実行部11によって更新される。

【0015】図4および図5はそれぞれ状態遷移図作成部8により自動合成されたプロセスP1およびP2の状態遷移図の例を示している。これらの図において、楕円形（例えばa）のシンボルがプロセスの状態、長方形（例えばb）のシンボルがタスク処理をそれぞれ示している。また図中例えばcの形状のシンボルが信号の受信、例えばdの形状のシンボルが信号の送信をそれぞれ示している。また送受信シンボルの中には「プロセス名（信号名）」という形式でその内容が記述されており、プロセス名が省略されているものは外部環境との通信を意味する。なお、gの形状のシンボルは条件分岐状態を示している。

【0016】さらに、例えばbのタスク処理シンボルに隣接して付加された“IT”は当該タスク処理に費やされる時間、例えば図4のeの受信シンボルに付加された“ $T_0=20T$ ”はタイムアウト時間をそれぞれ示しており、これらタスク処理時間およびタイムアウト時間はシミュレーション環境設定部10で設定される。

【0017】以上のシミュレーション環境および状態遷移図はシミュレーション時に同時に表示される。

【0018】次に本実施例のプロトコル・シミュレーション装置の動作を説明する。

【0019】図6はシステム全体の処理手順を示すフローチャートである。まずシミュレーション実行部11は、各プロセスの状態遷移図のシンボルを解釈しながらシンボルに対応したイベント処理を実行する（ステップ602）。但し、シミュレーション実行部11は、その前に各プロセスのイベントの待ち状態を調べ（ステップ601）、各プロセスのイベント待ち状態がすべて“初期状態（IDLE）”あるいは“内部信号の受信待ち（I）”でない場合に限ってこれを実行する。

【0020】図7は各プロセスでのイベント処理の手順を示すフローチャートである。各プロセスでのイベント処理は、個々のプロセスのタスク処理時間（ptime）がシミュレーションの全体タスク処理時間（ T_a ）と等しい場合に実行される（ステップ701）。

【0021】まずイベントの種類を調べる（ステップ702）。ここで“外部信号の受信”または“内部条件判断（状態遷移先の分岐）”のイベントを判定した場合、シミュレーション実行部11はユーザに分岐可能なすべての状態を表示して処理の選択を要求する（ステップ703）。この後、イベントを次に進め（ステップ704）、ステップ702に戻る。

【0022】またステップ702にて“内部信号の受信”のイベントを判定した場合、シミュレーション実行

10

20

30

40

50

部11は現在のFIFOキューの状態を調べ(ステップ705)、FIFOキューに正しい信号が保持されてことを判定すると、FIFOキューから当該信号を取り出す(ステップ706)。そしてイベントの待ち状態を“その他(else)”に設定する(ステップ707)。この後、イベントを次に進め(ステップ708)、ステップ701に戻る。また、FIFOキューが空であることを判定すると、シミュレーション実行部11はイベントの待ち状態を“内部信号の受信待ち(I)”に設定し(ステップ709)、正常終了とする(ステップ710)。さらにFIFOキューに受信待ち信号とは異なる信号が入っていることを判断すると、シミュレーション実行部11は未定義受信状態であることをユーザに知らせ(ステップ711)、異常終了とする(ステップ712)。

【0023】またステップ702にて“外部・内部信号の送信”のイベントを判定した場合、シミュレーション実行部11は送信先のプロセスに接続しているチャンネルに当該信号を送り(ステップ713)、イベントを次に進め(ステップ708)、ステップ701に戻る。

【0024】ここで、チャンネルに送信された信号は、図8に示すように、シミュレーション環境設定において当該チャンネルに設定された信号伝送時間が経過するまでこのチャンネルに格納され、設定時間の経過後(ステップ801)、チャンネルから信号を取り出してFIFOキューに渡す(ステップ802)。

【0025】またFIFOキューは、図9に示すように、シミュレーション環境設定において設定されたチャンネルの通信容量までの量の信号を受信可能であり、信号の数が通信容量を越えた場合(ステップ901)、チャンネルオーバーフローが発生したものを見なしてこの旨をユーザに知らせる(ステップ902)。

【0026】さらにステップ702で“タイマーのON”のイベントを判定した場合、シミュレーション実行部11はプロセス・マネージャーPMのタイマー属性を起動してタイマー設定を行い(ステップ714)、イベントを次に進めた後(ステップ708)、ステップ701に戻る。

【0027】またさらにステップ702で“タイマーのOFF”のイベントを判定した場合、シミュレーション実行部11は前記のタイマー設定を解除し(ステップ715)、イベントを次に進めた後(ステップ708)、ステップ701に戻る。

【0028】またステップ702で“タスクの処理”のイベントを判定した場合、シミュレーション実行部11はシミュレーション環境設定において設定されたタスク処理時間を消費して、プロセス・マネージャーPMの管理しているプロセスのタスク処理時間(ptime)をイベント処理時間分進める(ステップ716)。そしてイベントを次に進め(ステップ717)、正常終了とする(ス

テップ718)。

【0029】一方、図6のステップ601において、各プロセスの全ての待ち状態が“内部信号の受信待ち(I)”の場合は次のような処理を行う。

【0030】まず全てのチャンネルとFIFOキューの状態を調べると共にタイマー設定の有無を調べる(ステップ604)。この結果、全てのチャンネルとFIFOキューが空でしかもタイマーが未設定の場合は、デッドロック状態が発生しているものとしてユーザにこの旨を表示する(ステップ605)。そして異常終了とする(ステップ606)。

【0031】またその他の場合、つまり全ての待ち状態が“内部信号の受信待ち(I)”でしかもチャンネルあるいはFIFOキューに一つでも信号が存在する場合、あるいは全てのチャンネルとFIFOキューが空でもタイマーが設定されている場合は、全体のタスク処理時間Taおよびプロセスのタスク処理時間(ptime)をそれぞれ1タスク処理設定時間更新し(ステップ607、608)、さらにタイマーが設定されているプロセスについてはそのタイマー時間も1タスク処理設定時間更新する(ステップ609)。

【0032】この後、タイマー時間とタイムアウト時間とを比較して(ステップ610)、タイマー時間かタイムアウト時間を越えている場合は、非実行状態遷移の可能性をユーザに知らせる(ステップ611)。またタイマー時間かタイムアウト時間を越えていなければステップ601に戻る。

【0033】以上の手順で各プロセスの状態遷移図に従ってシミュレーションを実行し、各プロセスの状態が全て“初期状態(IDLE)”になったことをステップ601にて判定すると、シミュレーションを終了する(ステップ612)。

【0034】次にチャンネルオーバーフロー発生に関する動作について図10乃至図12を用いて説明する。図10はこの場合の表示例を示している。

【0035】チャンネルオーバーフローは、前述したように、チャンネルに送信された信号の数がチャンネルの通信容量を越えることで検出される。図10の例はチャンネルに“2”の通信容量が設定された前提において、当該チャンネルに接続されたFIFOキューに既に2つの信号が保持されているのにも関わらず、次の信号がチャンネルからFIFOキューに到達したため、チャンネルオーバーフローとなった例である。図11はこのチャンネルオーバーフローを引き起こす原因となった状態遷移図を示している。ここで、斜線により塗り潰されたシンボルが現在検証中であることを示している。この状態遷移図では、プロセスP2に“データを4Tの時間セーブする(save data)”という処理が存在し、これに起因してチャンネルオーバーフローが発生している。これをさらに分かりやすくしたものが図12である。この図から分かるよう

に、全体のタスク処理時間の“4T”の時刻で、チャンネルの通信容量を越えた3つ目の信号がFIFOキューに到着してしまっている。

【0036】次に非実行遷移状態発生の発生に関する動作について図13乃至図15を用いて説明する。図13はこの場合の表示例を示している。

【0037】非実行遷移状態の発生は、前述したように、プロセスのタイマー時間が設定タイムアウト時間を越えることで検出される。図14はこの非実行遷移状態を引き起こす原因となる状態遷移図、図15はこの状態遷移図に基づくプロセス間の処理手順を示している。ここで、プロセスP1には設定タイムアウト時間として“4T”が定義されている。ところが、プロセスP1とプロセスP2との間の各チャンネルP12、P21の信号伝送に“2T”、プロセスP2の処理に“1T”すなわち合計“5T”の時間がかかることから、プロセスP1では状態1においてタイムアウトとなり“+P2(ack)”や“+P2(nck)”は実行されない。

【0038】次にデッドロック状態発生に関する動作について図16乃至図18を用いて説明する。図16はこの場合の表示例を示している。

【0039】デッドロック状態の発生は、前述したように、各プロセスが“内部信号の待ち状態”にあり、かつプロセス間のチャンネルとFIFOキューが空でしかもタイマーが未設定の場合に判定される。図17はこのデッドロック状態を引き起こす原因となる状態遷移図、図18はこの状態遷移図に基づくプロセス間の処理手順を示している。この例では各プロセスP1、P2それぞれの“状態1”にてデッドロック状態が起きている。

【0040】最後に未定義受信状態発生に関する動作について図19乃至図21を用いて説明する。図19はこの場合の表示例を示している。

【0041】未定義受信状態の発生は、前述したように、FIFOキューに受信待ち信号とは異なる種類の信号が入っている場合に判定される。図20はこの未定義受信状態を引き起こす基となる状態遷移図、図21はこの状態遷移図に基づくプロセス間の処理手順を示している。この例では、プロセスP1では“+P2(ack)”、プロセスP2では“+P1(ack)”の信号を受信待ちしているものの、それぞれ送られてきた信号は“-P1(call)”“-P2(call)”であり、種類が異なるため、未定義受信状態と判定される。

【0042】したがって、本実施例の Protokol・シミュレーション装置によれば、Protokol設計を行う環境上でProtokolの動作を確認しながらProtokol設計を行うことができる。この結果、特にProtokol設計の重要な問題の一つであるタイムアウト処理を含む場合のProtokolを正確に設計することができ、Protokolの設計・検証作業を効率良く行うことができる。

【0043】また本実施例によれば、特別な実地環境や

模擬環境を要することなく、Protokolの動的な検証を自動的にかつProtokol設計者と対話的に進めることができ、通信システムの開発期間の大幅短縮、開発コストの低減化が達成される。

【0044】さらに本実施例によれば、チャンネルオーバーフロー状態、非実行遷移状態、デッドロック状態、未定義受信状態等のエラー表示をユーザに分かりやすく表示することができ、しかも現在検証中の状態遷移図も併せて表示することで、エラー発生原因の解明も容易になると言う利点を有している。

【0045】

【発明の効果】以上説明したように本発明の Protokol・シミュレーション装置によれば、特別な実地環境あるいは模擬環境を要することなく、通信Protokolの動的なシミュレーションを自動的に行うことができ、Protokol検証作業の大幅な効率向上を図ることができる。

【図面の簡単な説明】

【図1】本発明に係る一実施例の Protokol・シミュレーション装置のシステム構成を示すブロック図である。

【図2】プロセス関連図記述部により記述された通信システムのプロセス関連図の例を示す図である。

【図3】図2のプロセス関連図に対してシミュレーション環境設定部で設定したシミュレーション環境を示す図である。

【図4】状態遷移図作成部により合成されたプロセスP1の状態遷移図の例を示す図である。

【図5】状態遷移図作成部により合成されたプロセスP2の状態遷移図の例を示す図である。

【図6】図1の Protokol・シミュレーション装置におけるシステム全体の処理手順を示すフローチャートである。

【図7】図6における各プロセスでのイベント処理の手順を示すフローチャートである。

【図8】チャンネルの処理手順を示すフローチャートである。

【図9】FIFOキューの処理手順を示すフローチャートである。

【図10】チャンネルオーバーフローが発生したときの検証結果表示を示す図である。

【図11】図10のチャンネルオーバーフローを引き起こす状態遷移図を示す図である。

【図12】図11の状態遷移図に基づくプロセス間の処理手順を示す図である。

【図13】非実行遷移状態が発生したときの検証結果表示を示す図である。

【図14】図13の非実行遷移状態を引き起こす状態遷移図を示す図である。

【図15】図14の状態遷移図に基づくプロセス間の処理手順を示す図である。

【図16】デッドロック状態が発生したときの検証結果

表示を示す図である。

【図17】図16のデッドロック状態を引き起こす状態遷移図を示す図である。

【図18】図17の状態遷移図に基づくプロセス間の処理手順を示す図である。

【図19】未定義受信状態が発生したときの検証結果表示を示す図である。

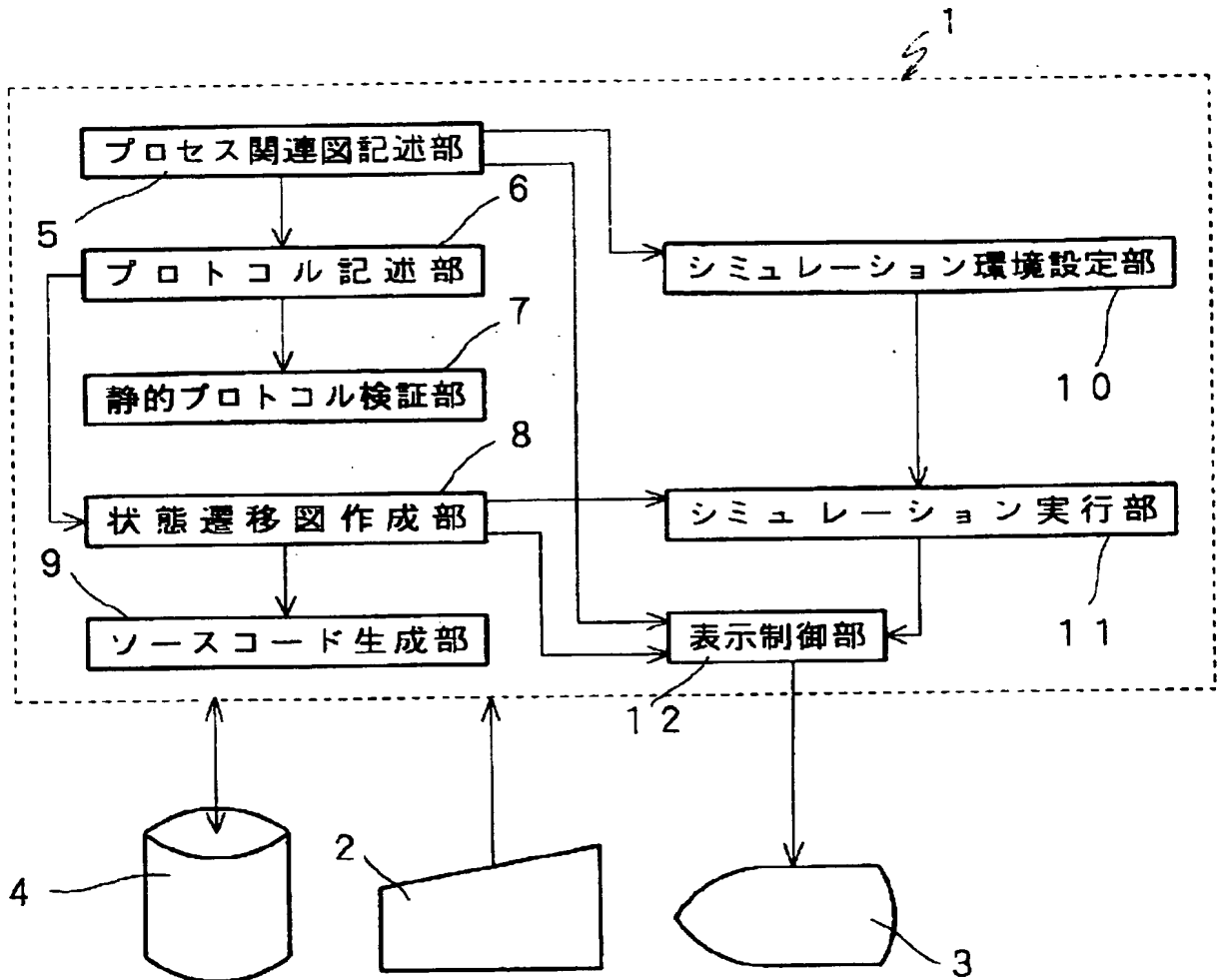
【図20】図19の未定義受信状態を引き起こす状態遷移図を示す図である。

【図21】図20の状態遷移図に基づくプロセス間の処理手順を示す図である。

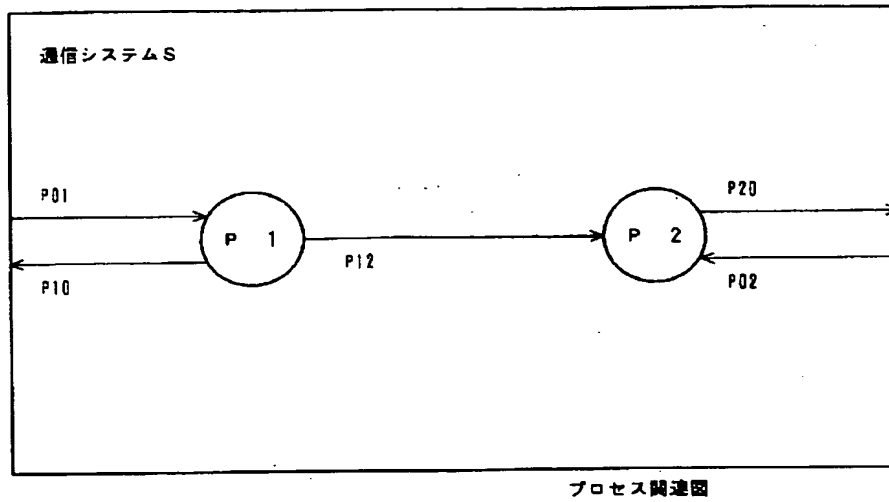
【符号の説明】

1…装置本体、2…入力装置、3…表示装置、4…外部記憶装置、5…プロセス関連図記述部、6…プロトコル記述部、7…静的プロトコル検証部、8…状態遷移図作成部、9…ソースコード生成部、10…シミュレーション環境設定部、11…シミュレーション実行部、12…表示制御部。

【図1】

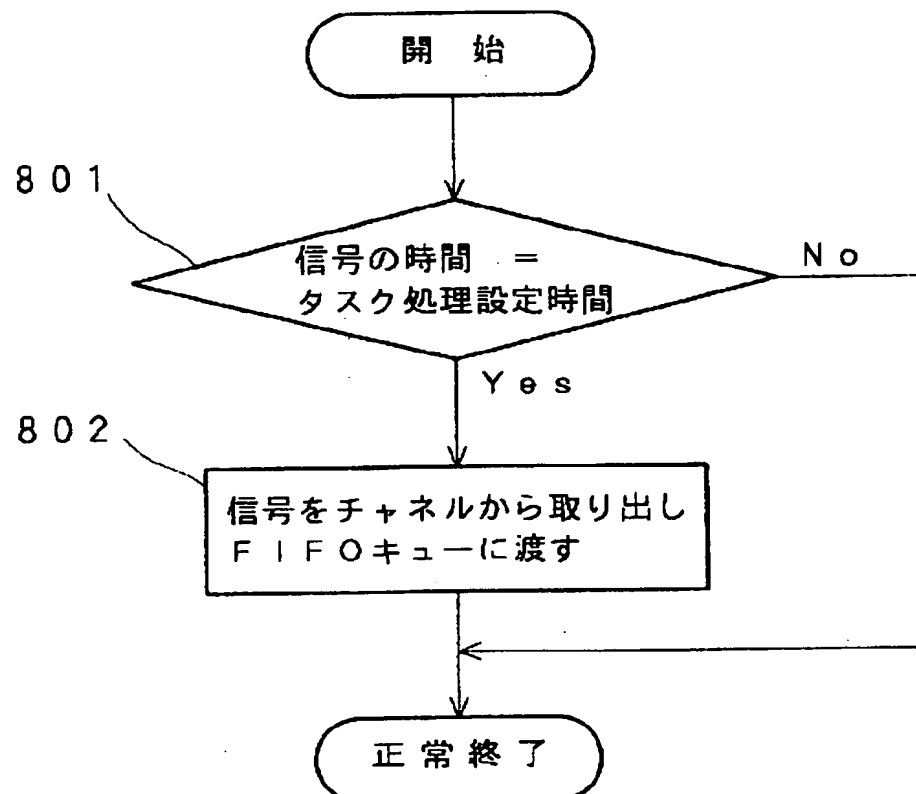


【図2】

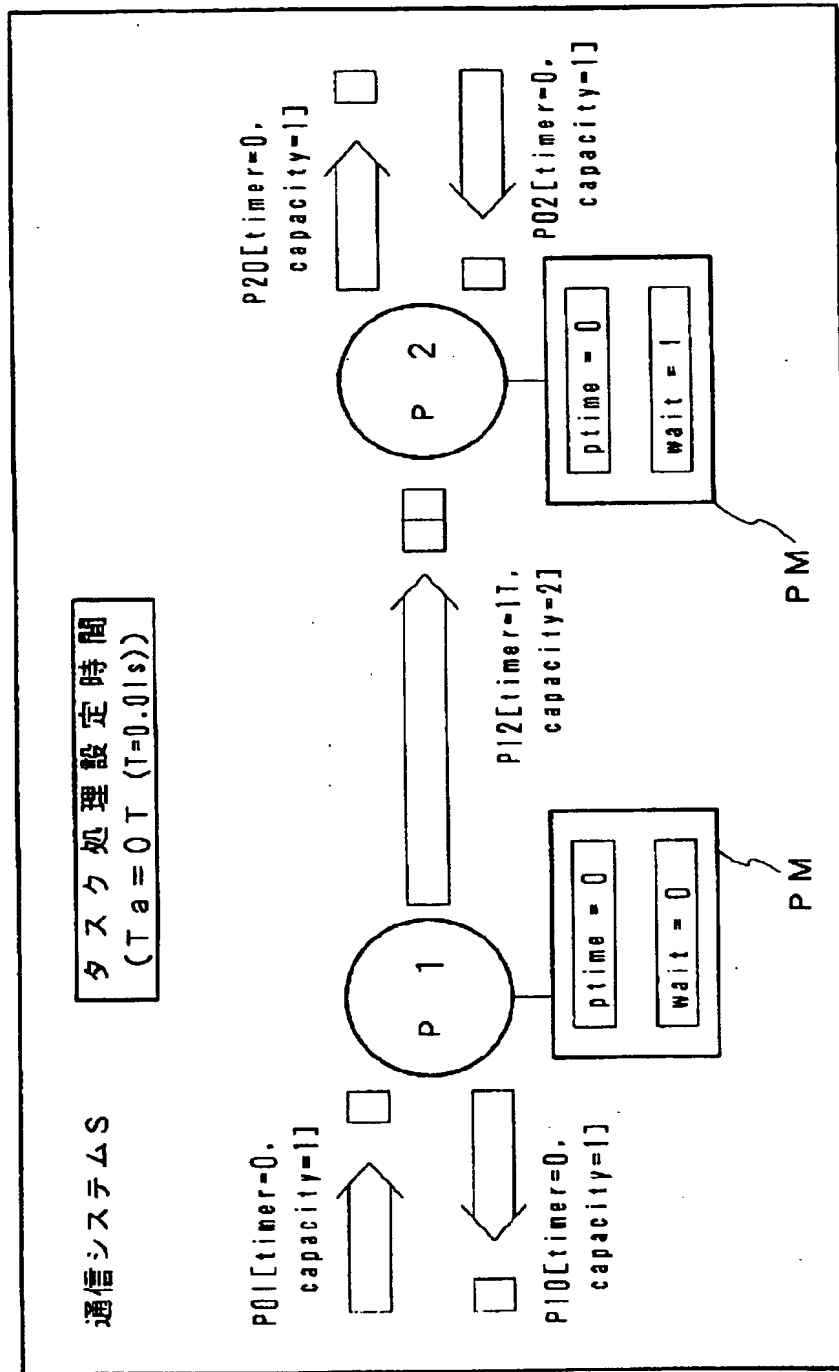


【図8】

チャネルの処理手順

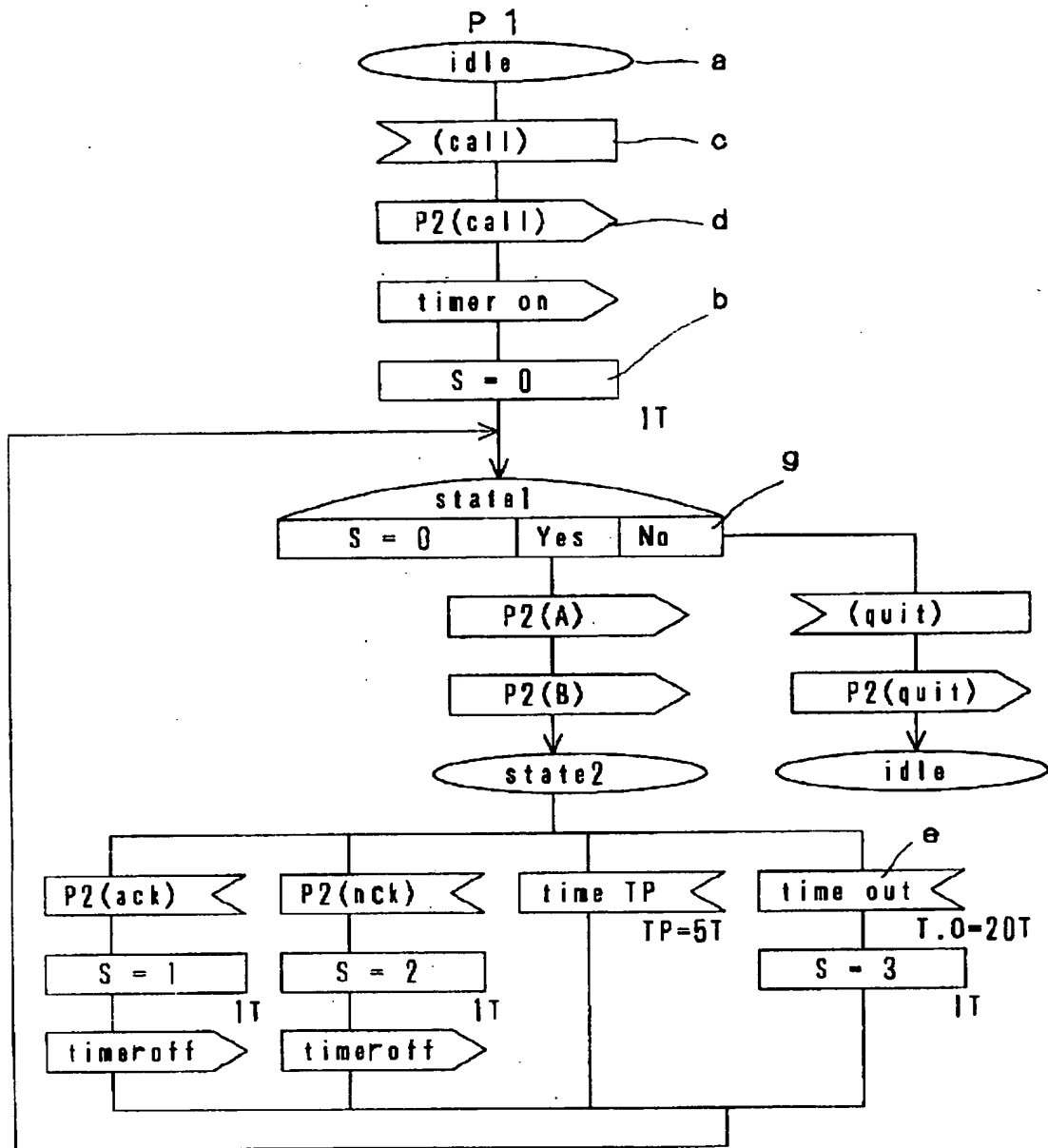


【図3】



シミュレーション環境

【図4】

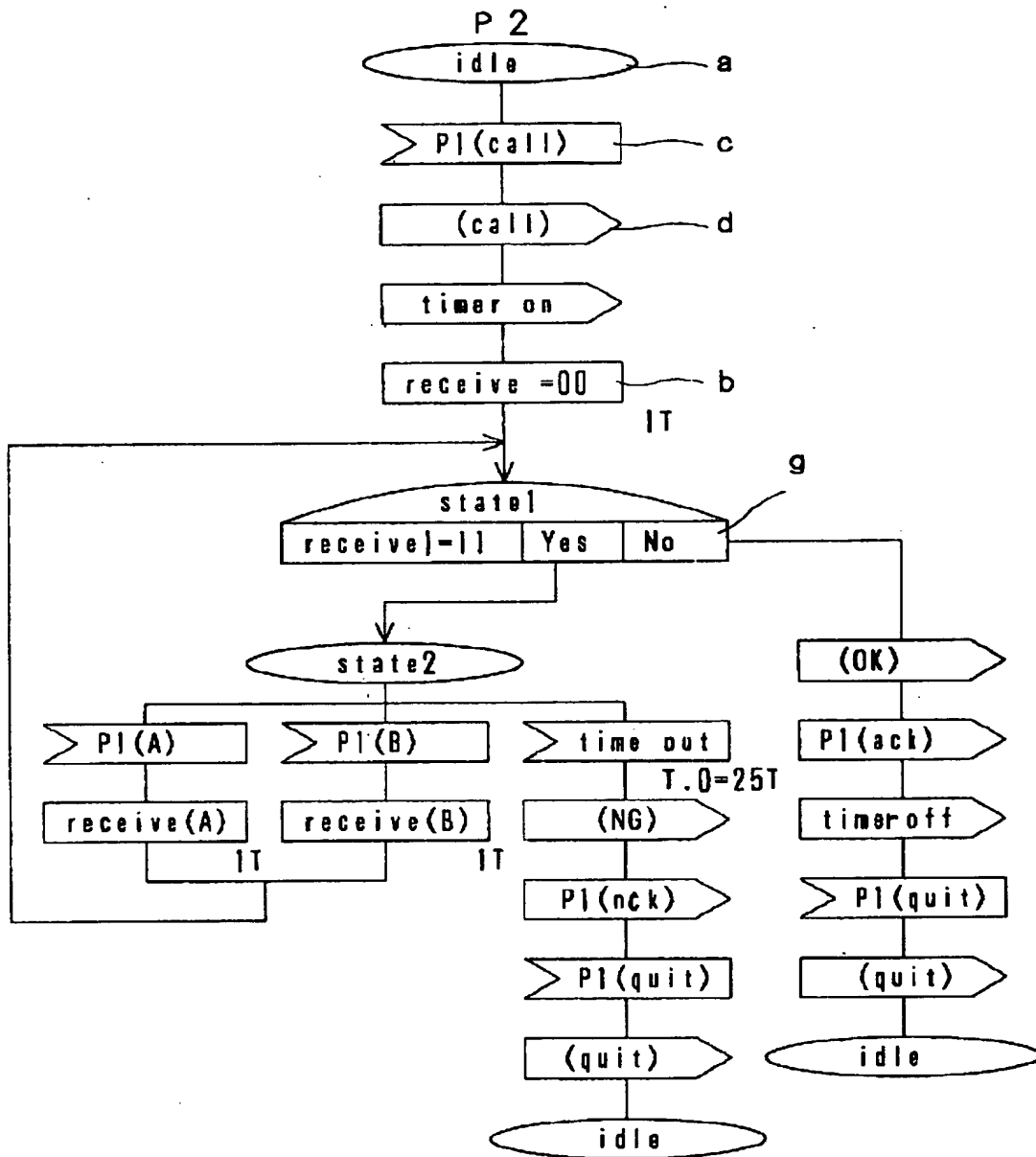


【図18】

デッド ロック状態の処理手順

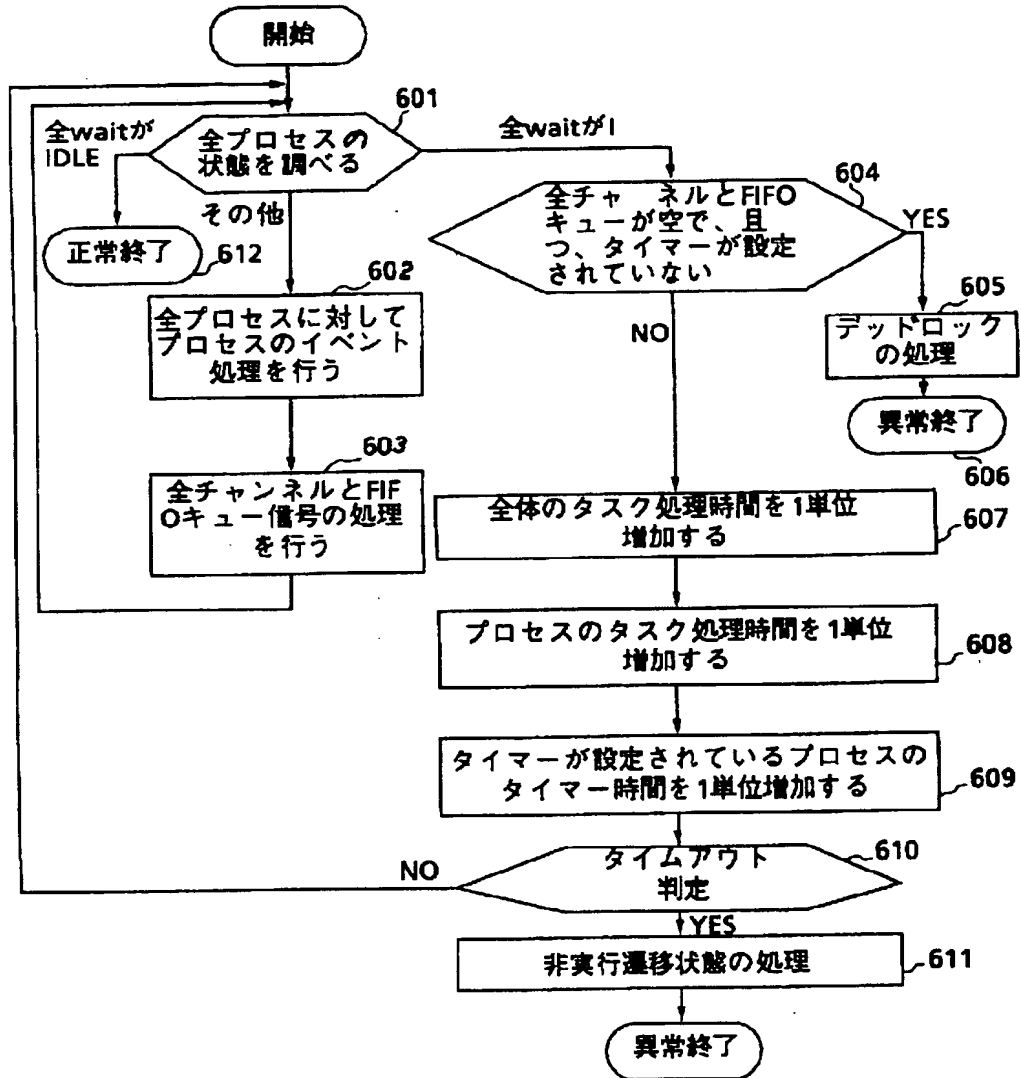
全体 (タスク処理)	P.1 (ptime)	P.1 2	P.2 1	P.2 (ptime)
0 T	0 T + (call) - P2(call)	call		0 T
1 T	1 T			1 T + P1(call)

【図5】



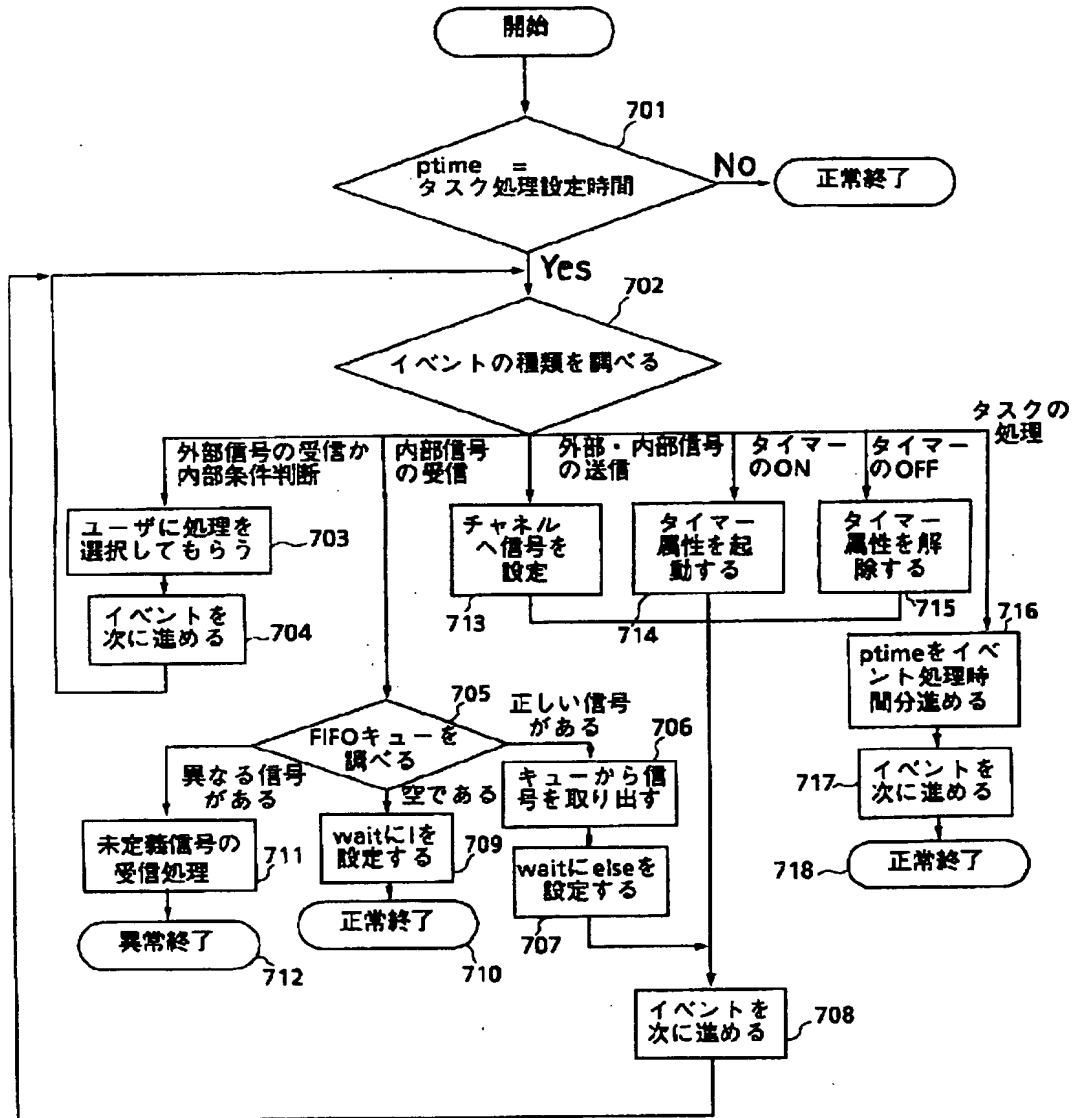
【図6】

システム全体の処理手順



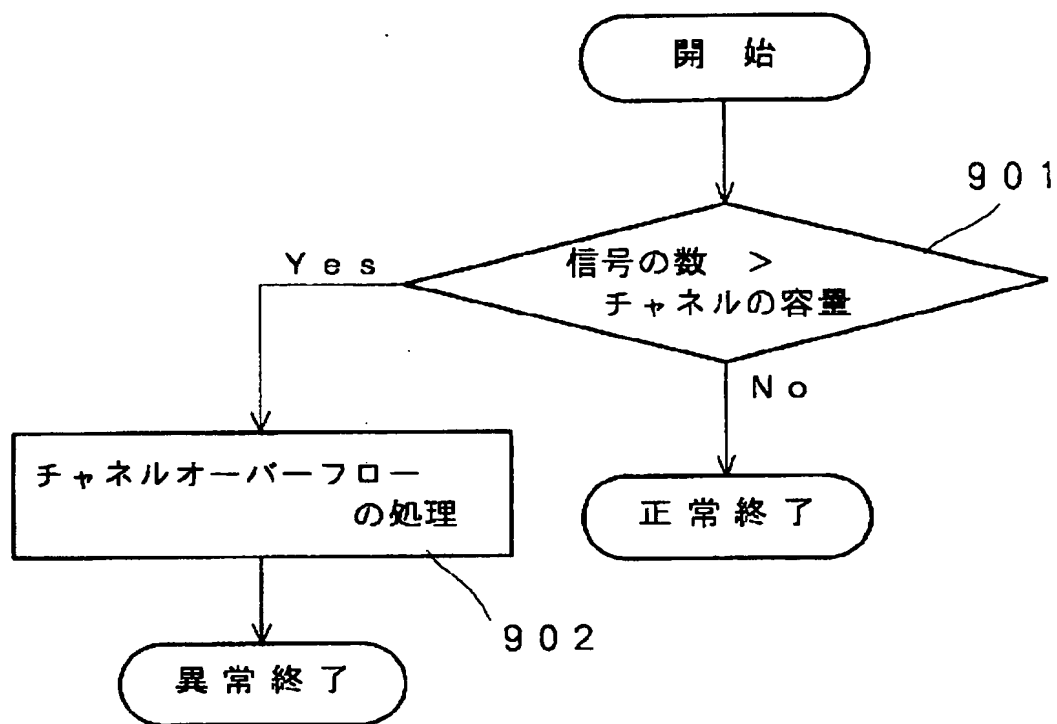
【図7】

各プロセスのイベント処理手順



【図9】

F I F O キューの処理手順

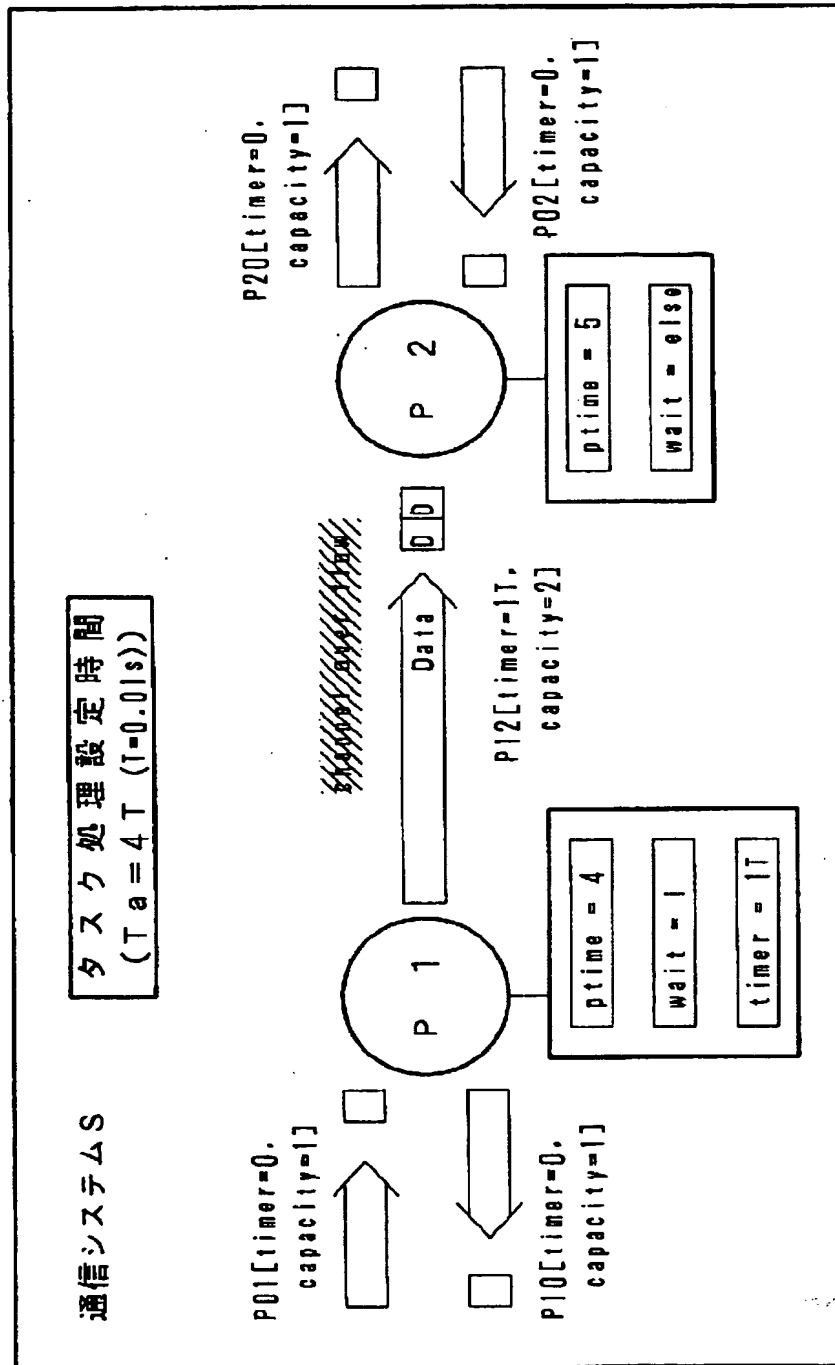


【図15】

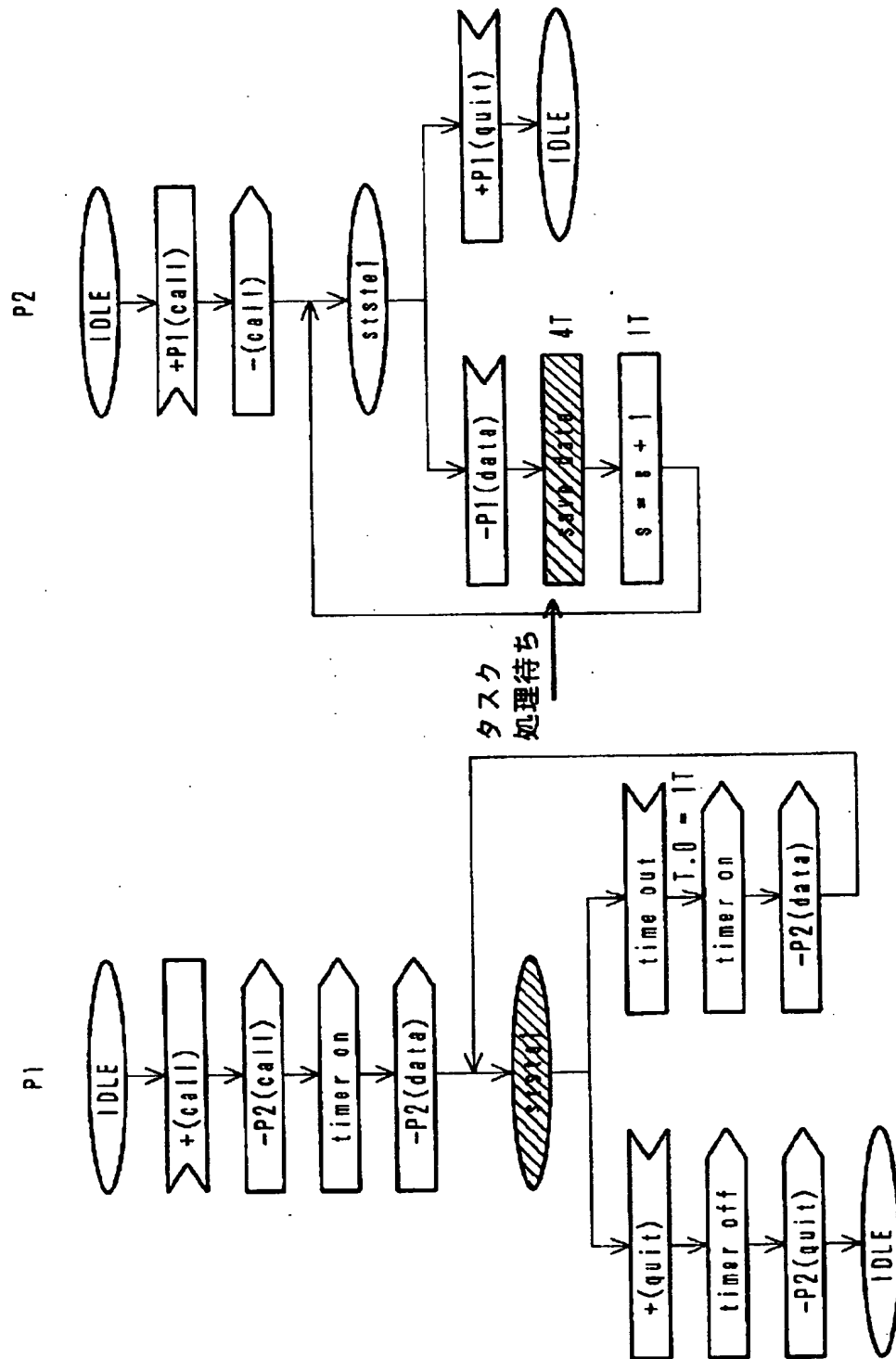
非実行遷移状態の処理手順

全体 (タスク処理)	P1 (ptime)	P12	P21	P2 (ptime)
0 T	0 T +(call) -P2(call) timer on	call		0 T
1 T	1 T	call		1 T
2 T	2 T			2 T +P1(call) -(call) check
3 T	3 T			3 T -P1(ack)
4 T	4 T time out exit		ack	4 T

【図10】



【図11】



【図12】

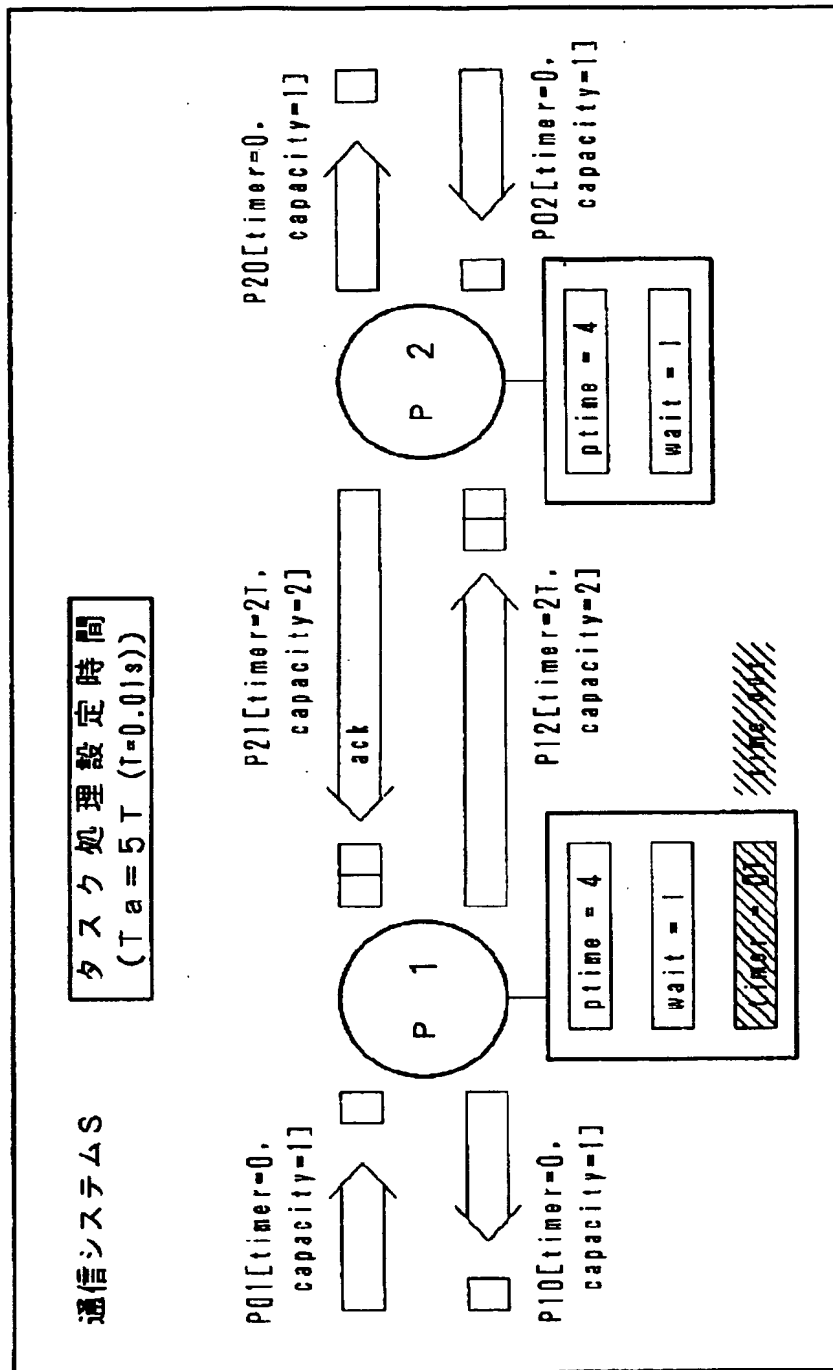
全体 (タスク処理)	P 1 (ptime)	P 1 2	F I F O	P 2 (ptime)
0 T ↓	0 T +(call) -P2(call) timer on -P2(data)	call data		0 T
1 T	1 T time out timer on -P2(data)	data		1 T +P1(call) -(call) +P1(data) save data
2 T	2 T time out timer on -P2(data)	data	data	5 T
3 T	3 T time out timer on -P2(data)	data	data, data	5 T
4 T	4 T time out timer on -P2(data)	data	data, data, data	5 T

【図21】

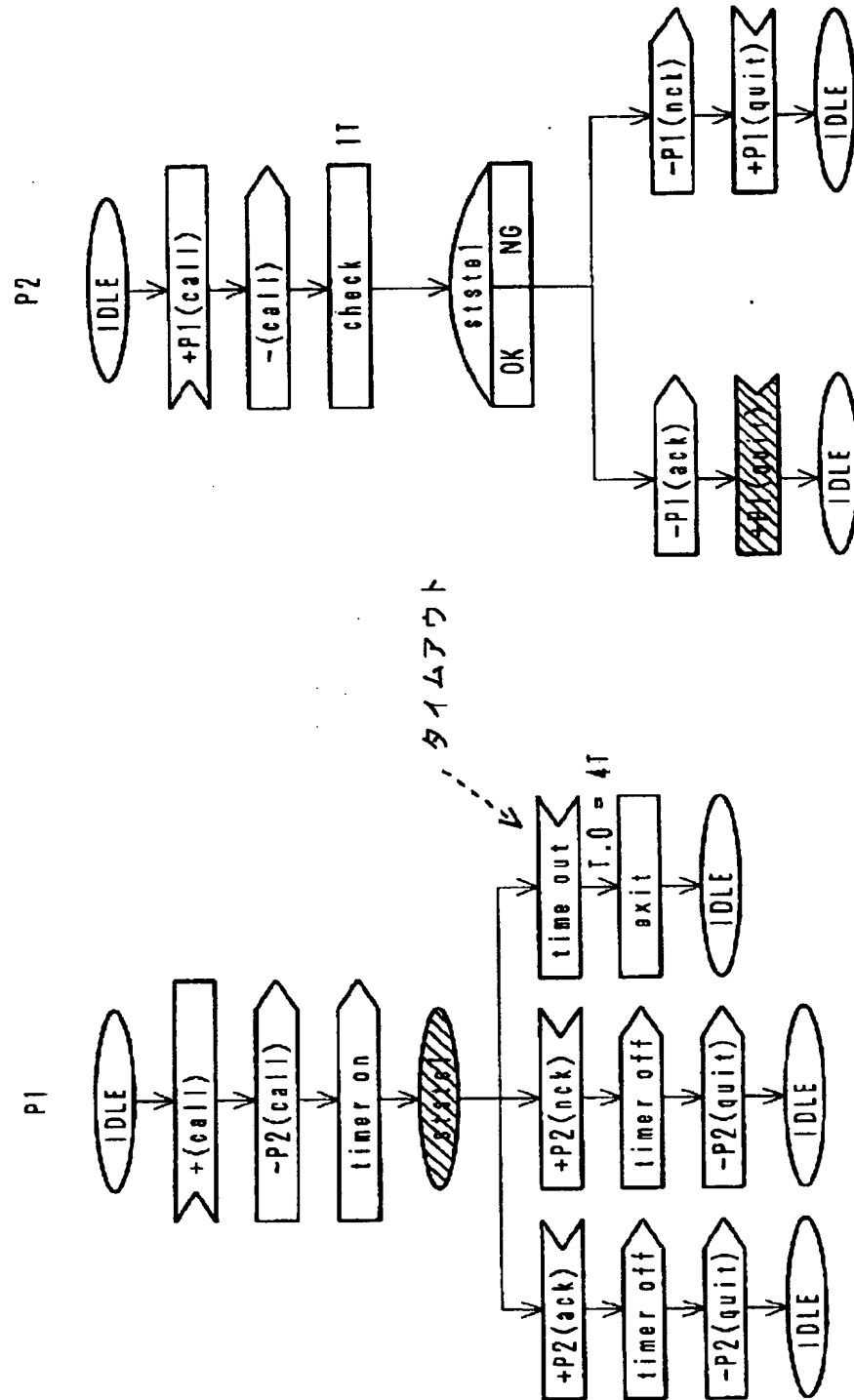
未定義受信状態の処理手順

全体 (タスク処理)	P 1 (ptime)	P 1 2	P 2 1	P 2 (ptime)
0 T	0 T +(call) -P2(call)	call		0 T +(call) -P1(call)
			call	
1 T	1 T +P2(call) を受けられない			1 T +P1(call) を受けられない

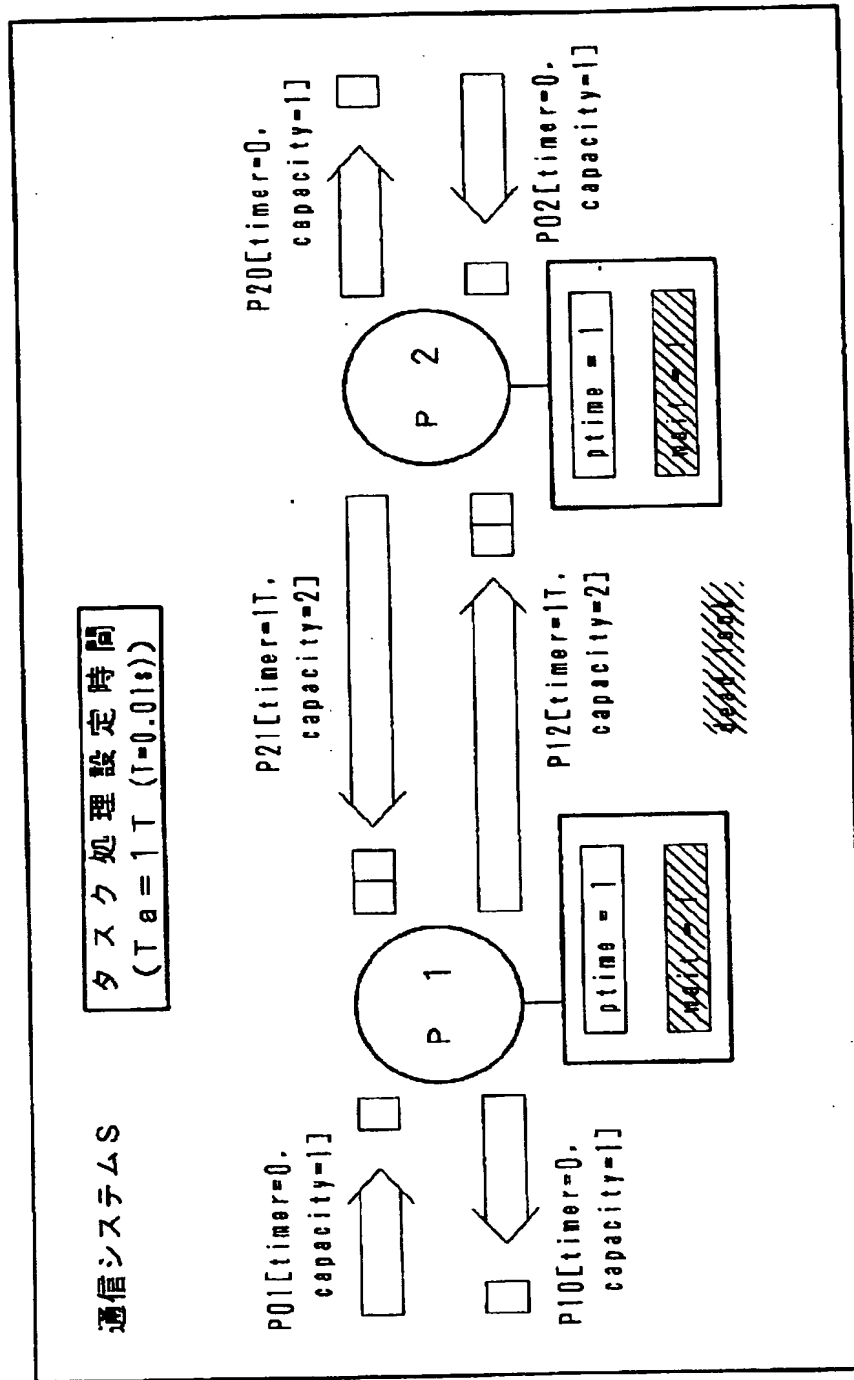
【図13】



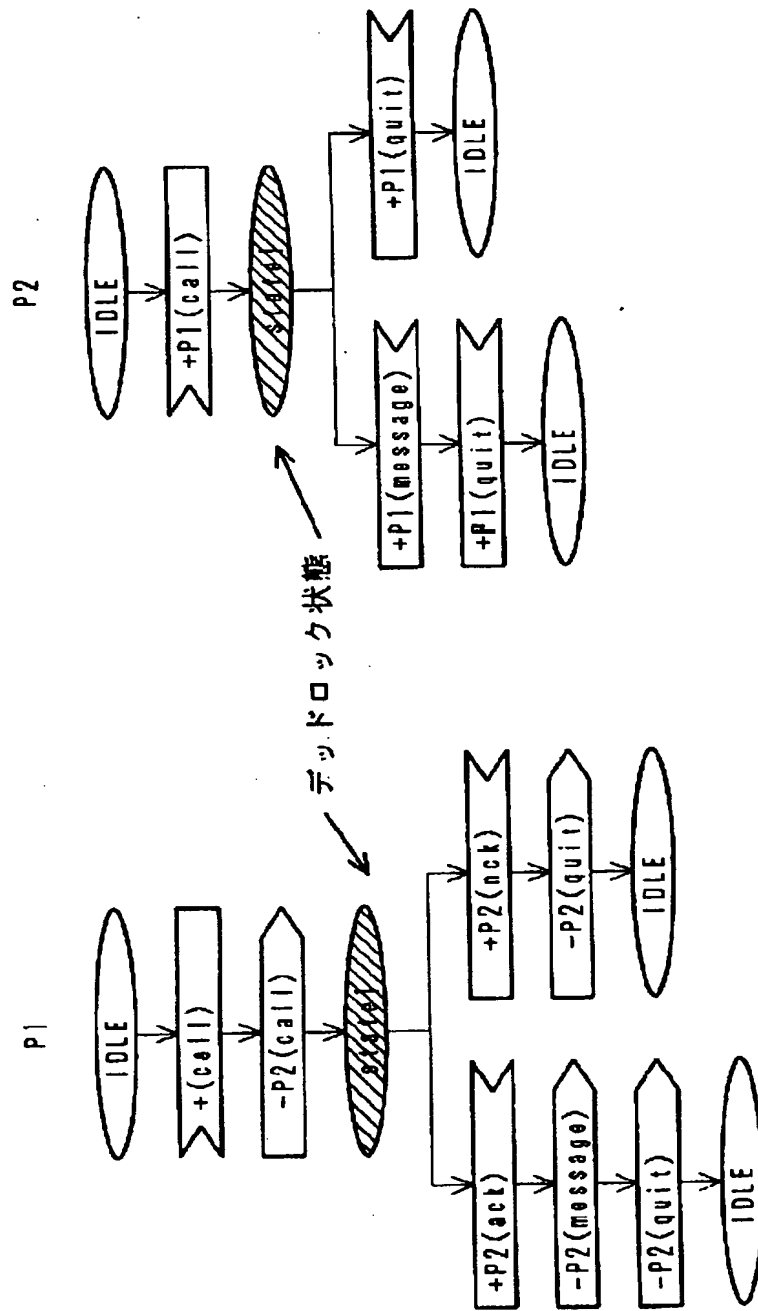
【図14】



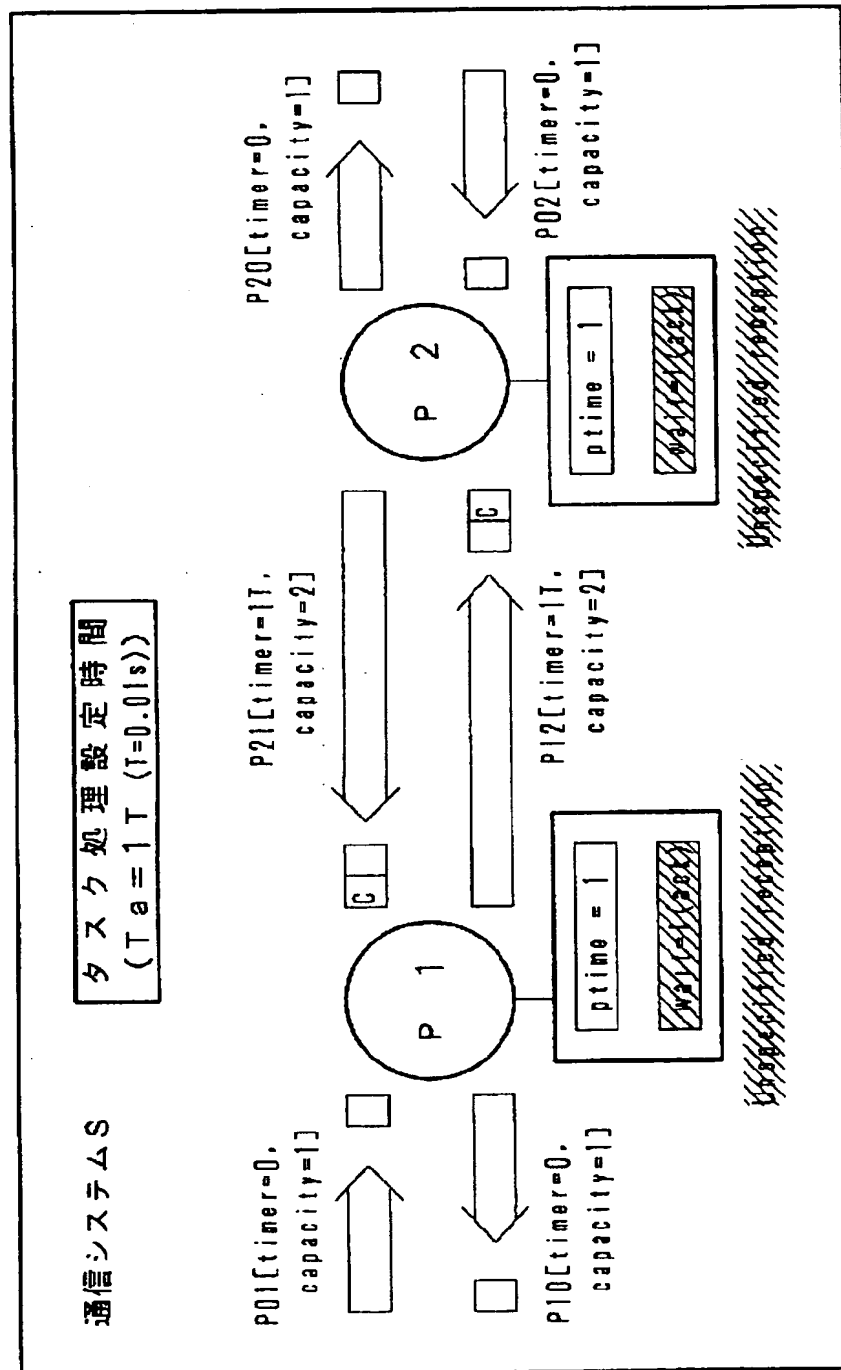
【図16】



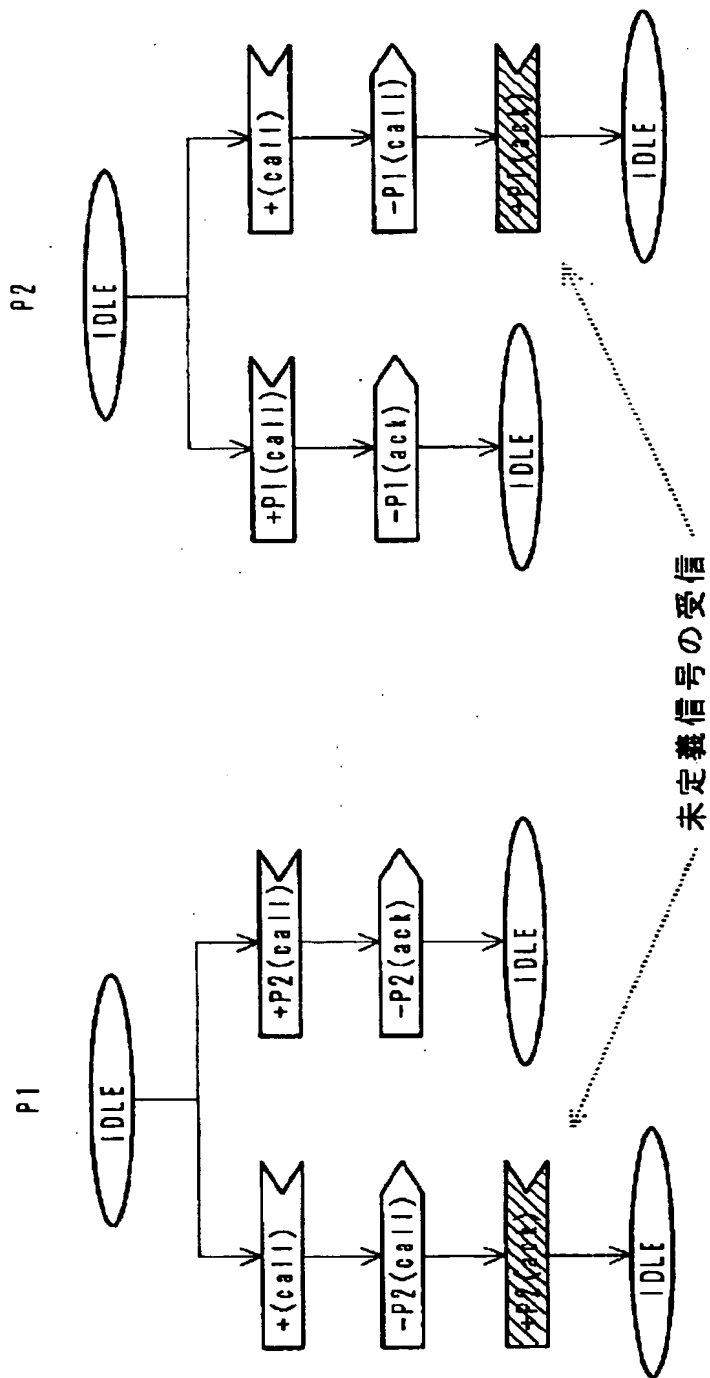
【図17】



【図19】



【図20】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☒ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.